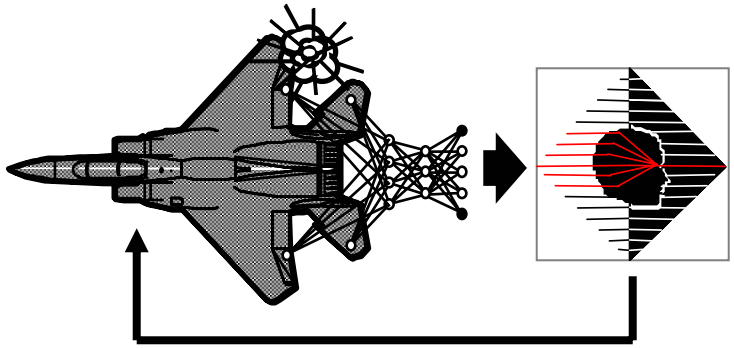
Intelligent Control Approaches for UAVs



K. KrishnaKumar
NeuroEngineering Laboratory
NASA Ames Research Center

Presented at UAV-MMNT03

Presentation Outline

- Intelligent Control Background
- Intelligent Flight Control Research @ NASA Ames

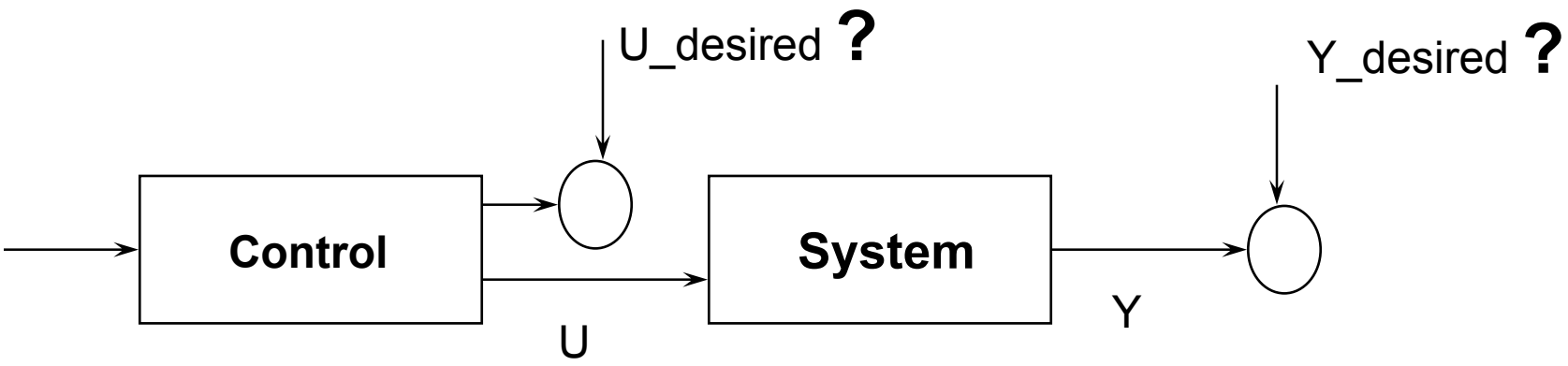
- Intelligent Control Background
  - What are intelligent systems
  - What is intelligent control
  - Intelligent control architectures

Defining Intelligent Systems

- An Intelligent System is one that exhibits any of the following traits:
  - ✓ Learning
  - ✓ Adaptability
  - ✓ Robustness across problem domains
  - ✓ Improving efficiency (over time and/or space)
  - ✓ Information compression (data to knowledge)
  - ✓ Extrapolated reasoning

IS is seen as Rationalistic AI: Intelligence for doing the right thing

Intelligent Control



Two Error Signals are needed:

1. System Performance Error Signal
2. Control Error Signal

Questions

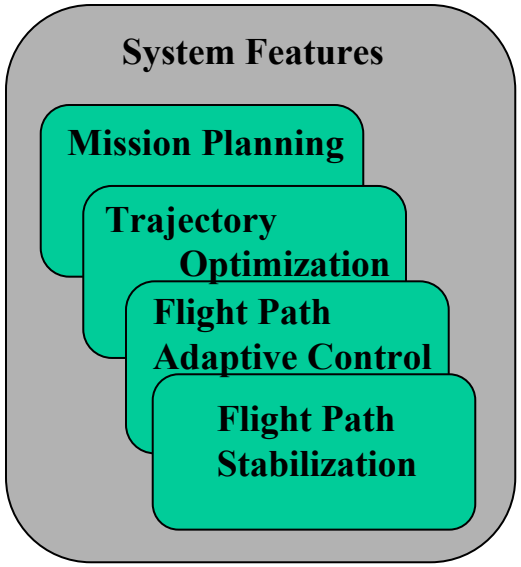
How do we say that one controller is more intelligent than the other?

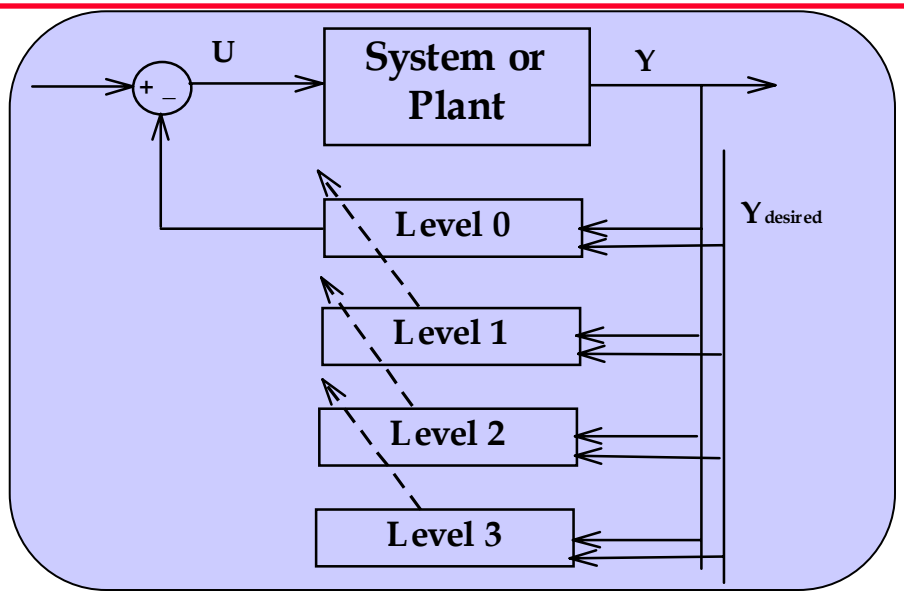
Can the intelligence be improved?

Can intelligence be measured?

Answer : Levels of Intelligent Control

Levels of Intelligent Control



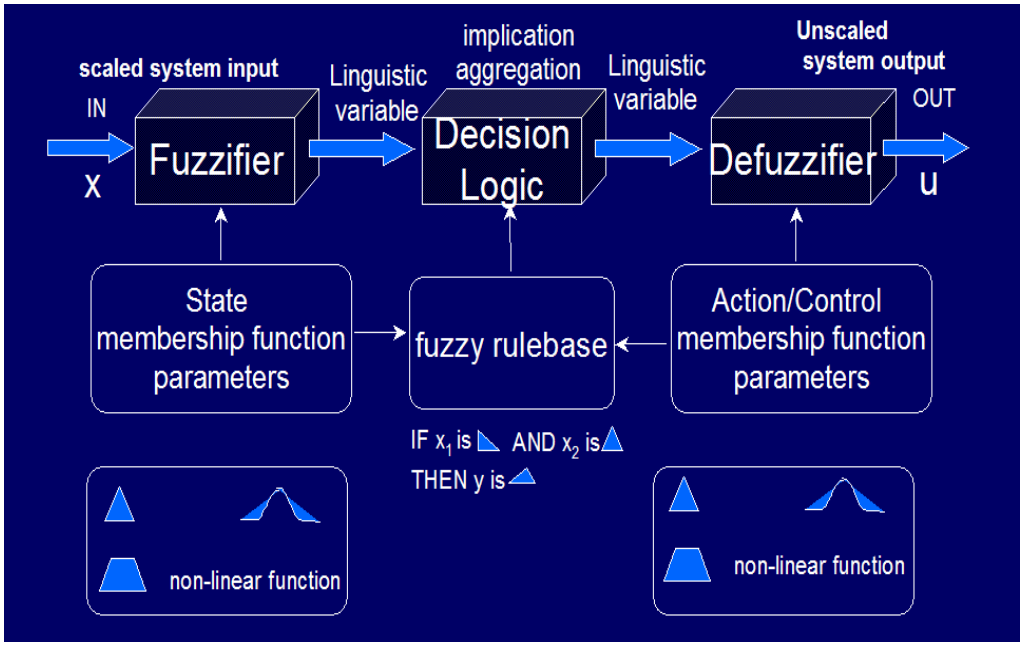


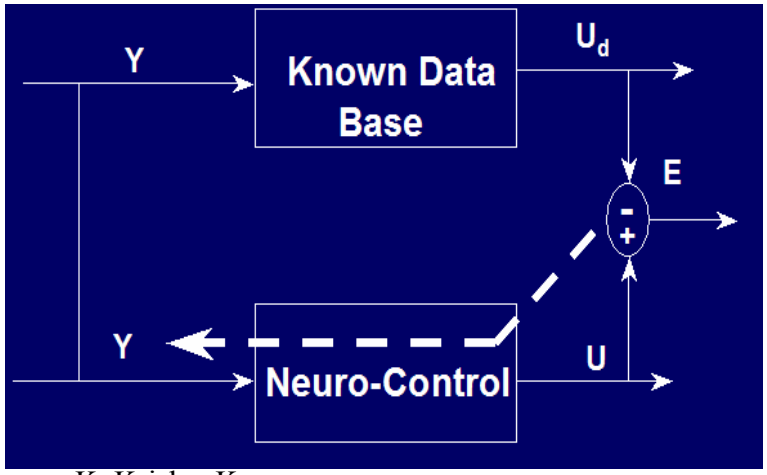
Lev	Self improvement of:	Description
0	Tracking Error (TE)	Robust Feedback control (Error tends to zero).
1	TE + Control Parameters (CP)	Robust feedback control with adaptive control parameters (error tends to zero for non-nominal operations; feedback control is self improving).
2	TE + CP + Performance Measure (PM)	Robust, adaptive feedback control that minimizes or maximizes a utility function over time (error tends to zero and a measure of performance is optimized).
3	TE+CP+PM+ Planning Function	Level 2 + the ability to plan ahead of time for uncertain situations, simulate, and model uncertainties.

Levels

➡ Level 0: Robust stabilization

- Gain Scheduling
- Supervised neuro-control
- Fuzzy control
- Mimic a controller
- Implicit Control

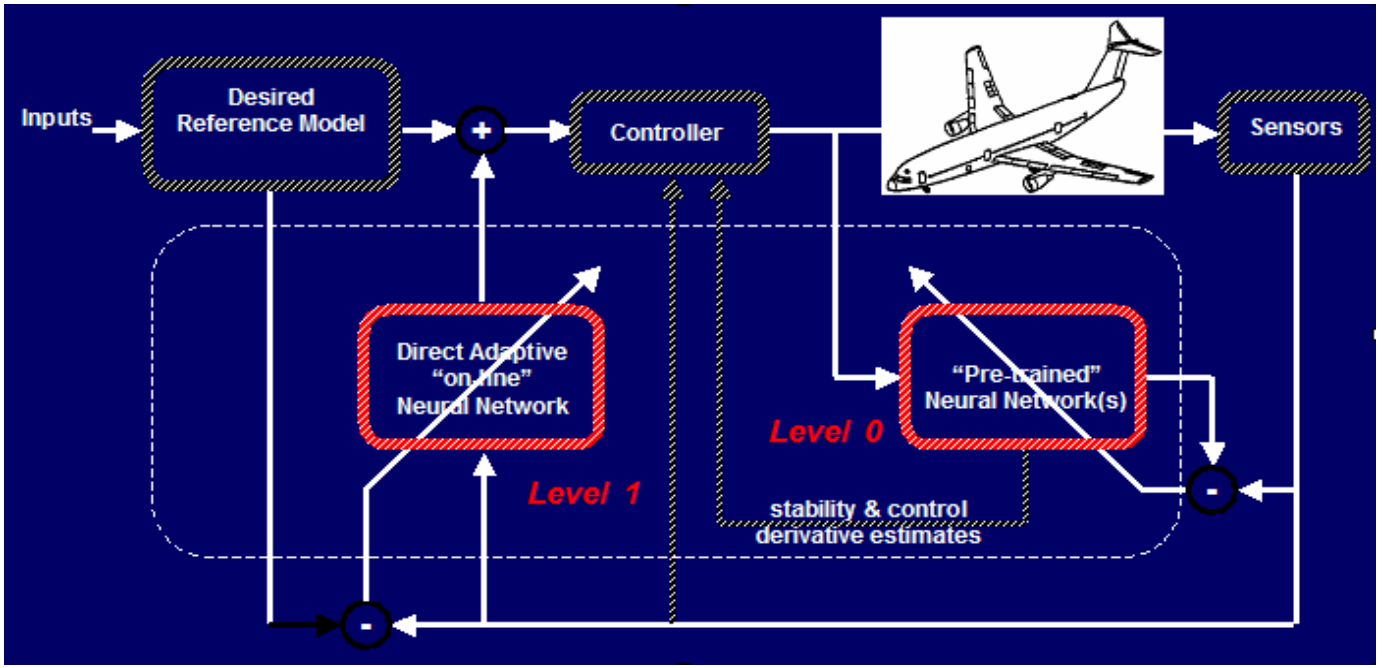




Levels

➡ Level 1: Adaptive Control

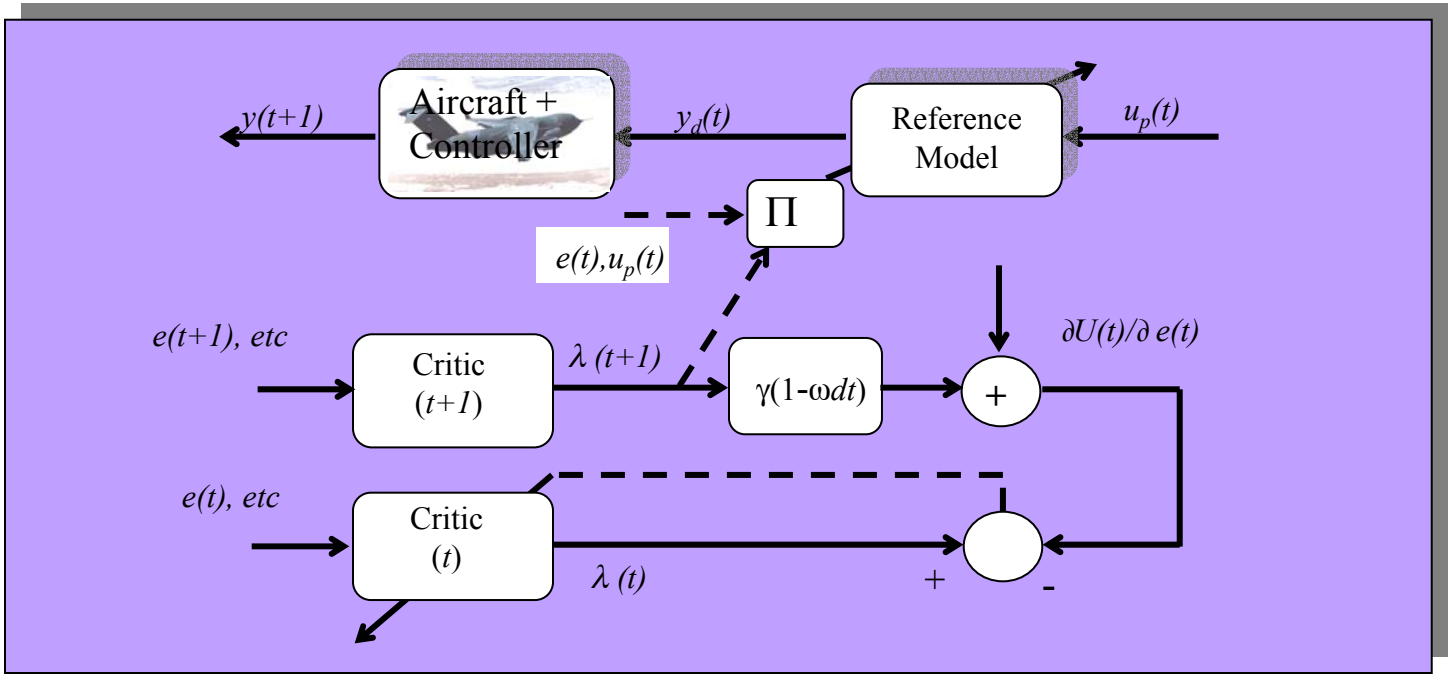
- Learn Systems and Controller Parameters
- Neural adaptive Control
- Adaptive inverse Control
- Approximate Controller error signal



Levels

➡ Level 2: Optimal Control

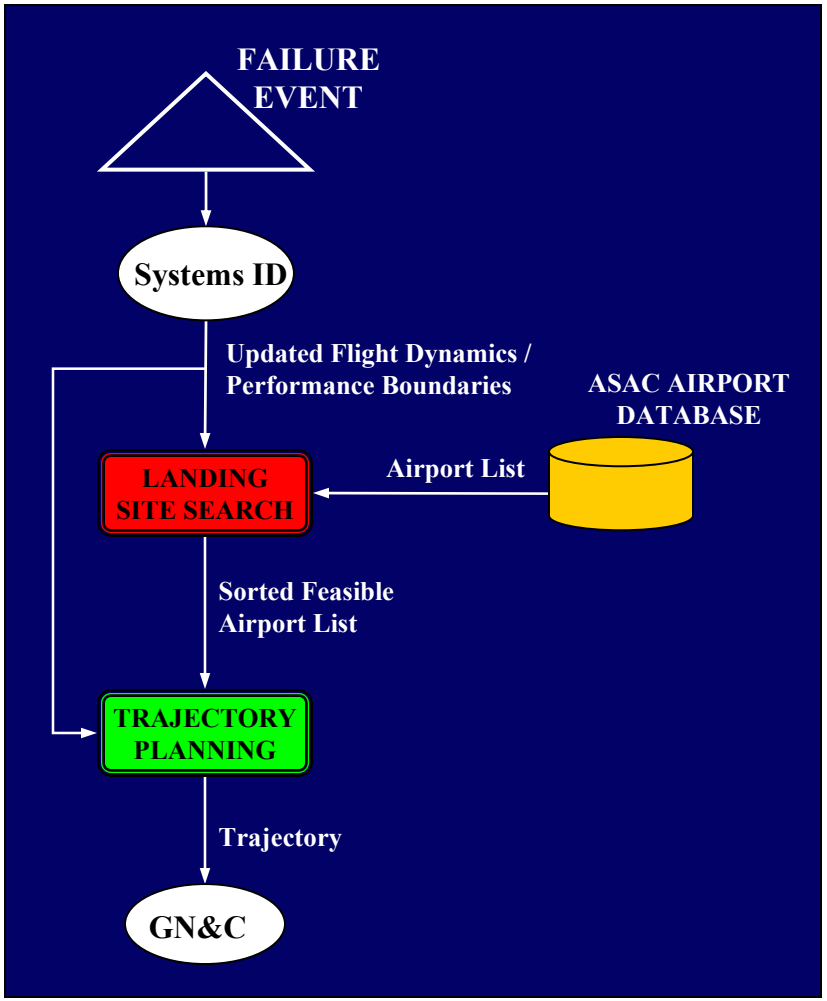
- **Reinforcement Learning**
- **Control Allocation**
- **Dynamic programming**
- **Linear Adaptive Critics**
- **Non-linear Adaptive Critics**



Levels

➡ Level 3: Planning Control (More AI-like)

- Strategic Planning
- Strategic search
- Mission Planning
- HTN: hierarchical task network
- Production-based cognitive architectures
- Decision-theoretic (MMDP)
- Etc..



NASA Ames
Intelligent Flight Control Applications

Manned Aircraft Objectives

Develop flight control technologies that can automatically compensate for problems or failures when they occur

Develop these technologies and capabilities in a generic sense so that they can be applied to different vehicle classes

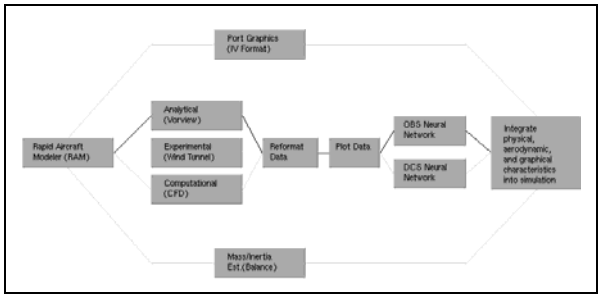
Application Platforms

- B 757 class aircraft – Simulation only
- F-15 – In Flight test
- C-17 – Flight tests in 2004

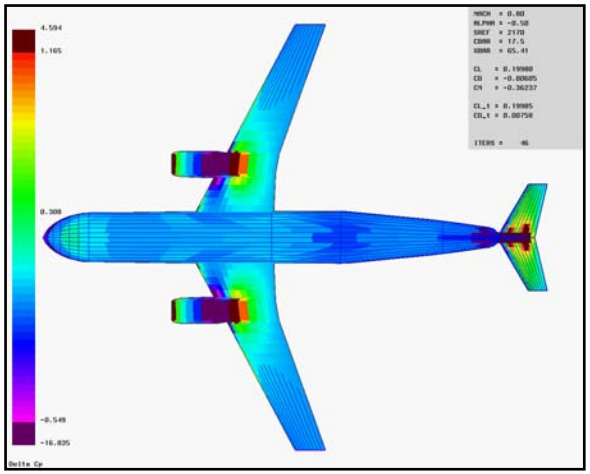
Pre-Trained Neural Networks

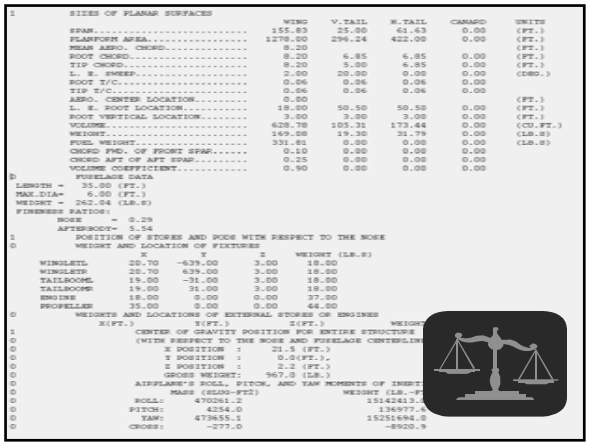
Ames Research Center

Step 1
Integrated Vehicle Modeling Env.
Rapid Aircraft Modeler (RAM)

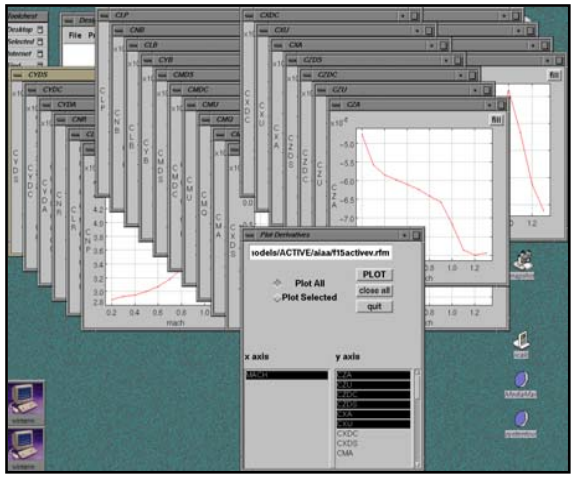


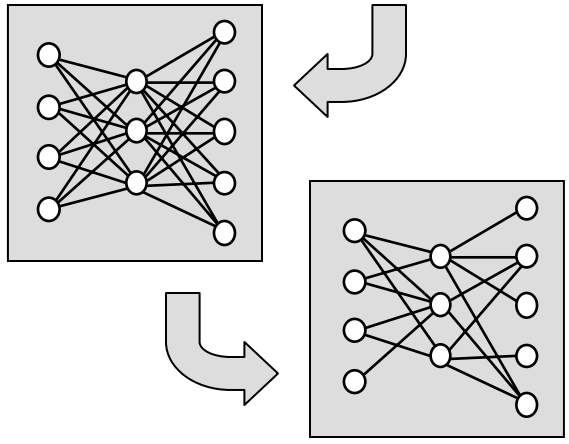
Step 2
Vortex Lattice Code (VORVIEW)
Mass/Inertia Estimates (Balance)





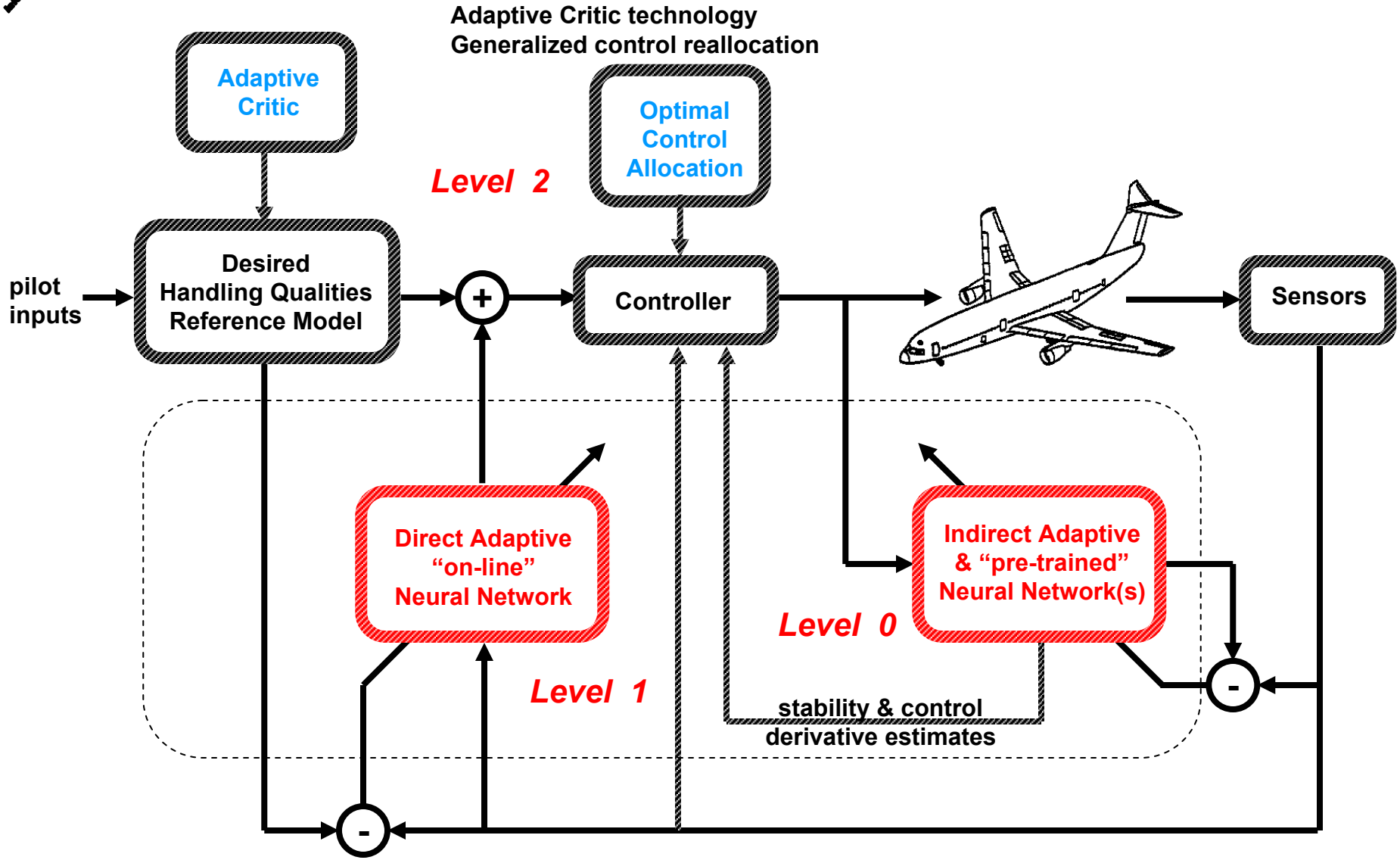
Step 3
Levenberg Marquardt neural net Optimal Pruning Algorithm





K. KrishnaKumar

Neural Flight Control Architectures



Level 1 Adaptive Control Equations

Plant: $\ddot{x} = f(\dot{x}, x, \delta)$

Linear approximation: $\ddot{x} \cong A\dot{x} + B\,\delta$

Control law design: $\delta = B^{-1}(\upsilon - A\dot{x})$

Closed loop: $\ddot{x} = f(\dot{x}, x, \delta) = \upsilon + \tilde{f}$

Inversion error: $\tilde{f} = f - \upsilon = f(\dot{x}, x, \delta) - (A\dot{x} + B\,\delta)$

Level 1 Control Equations

Provide compensation for the inversion error by design of υ

$$\upsilon = \upsilon_0 - \upsilon_{AD}$$

υ_0 is designed as output of a linear controller, e.g. "PI" control.

$$\upsilon_0 = \ddot{x}_c + K_p(\dot{x}_c - \dot{x}) + K_I(x_c - x);$$

υ_{AD} is the adaptive control

Level 1 Control Equations

Rewrite

$$\tilde{x} = (x_C - x) \quad \textbf{and} \quad e = \begin{bmatrix} \tilde{x} \\ \dot{\tilde{x}} \end{bmatrix}$$

With $a = \begin{bmatrix} 0 & 1 \\ -K_D & -K_P \end{bmatrix}$ **and** $b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

We have the tracking error dynamics as: $\dot{e} = a\ e + b(\upsilon_{AD} - \tilde{f})$

Neural Network Input Map:

$\upsilon_{AD} = W^T \mathcal{S}(x, V)$ **for sigma-pi NN**

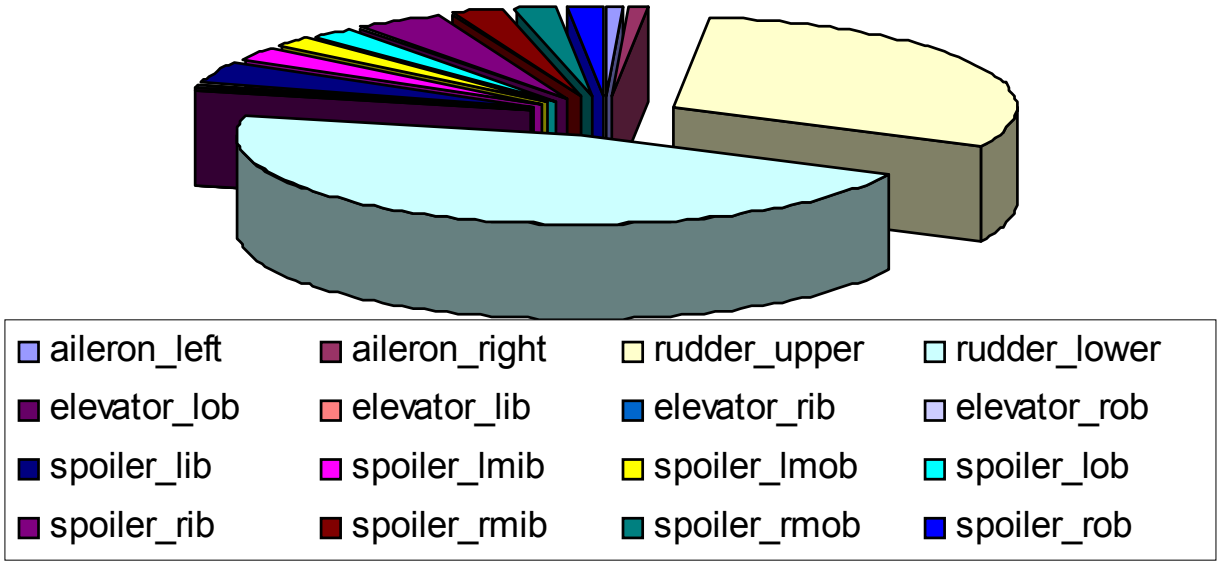
$\upsilon_{AD} = W^T \Phi(x, V)$ **for RBF NN**

Level 2: Optimal Control Allocation

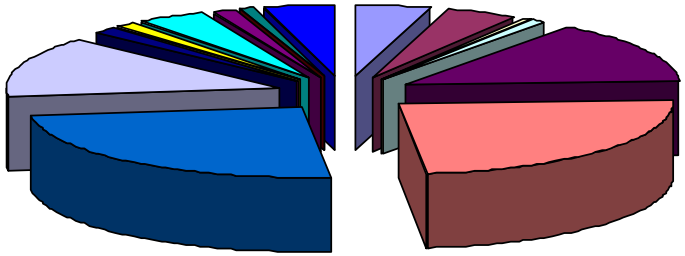
- When to allocate?
  - Control limit violation
  - Rate saturation
  - Control failure
- How to allocate?
  - Optimal allocation using Linear Programming
    - Conventional hierarchy
    - Best available hierarchy

Example Aerodynamic Control Authority

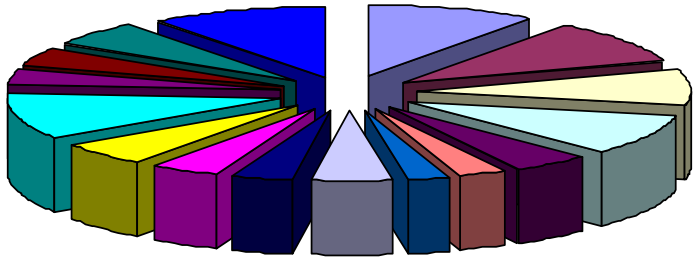
Directional Authority



Pitch Conrol Authority



Roll Control Authority



Linear Programming Formulation

Dynamic System is defined as

$$\left[\dot{X}\right] = f(X) + [B][u] + f_{trim}$$

Let us write $[B][u]$ as

$$\begin{bmatrix} B_{UU} & B_{UL} \\ B_{LU} & B_{LL} \end{bmatrix} \begin{bmatrix} u_U \\ u_L + \Delta u_L \end{bmatrix} = \begin{bmatrix} B_{UU} & B_{UL} \\ B_{LU} & B_{LL} \end{bmatrix} \begin{bmatrix} u_U \\ u_L \end{bmatrix} + \begin{bmatrix} B_{UU} & B_{UL} \\ B_{LU} & B_{LL} \end{bmatrix} \begin{bmatrix} 0 \\ \Delta u_L \end{bmatrix}$$

---

u_U = Unlimited Control Vector from Dynamic Inverse

$u_L + \Delta u_L$ = Limited Control Vector from Dynamic Inverse

L P Formulation (cont'd)

What we need is help from Unlimited Control

$$\begin{bmatrix} B_{UU}\Delta u_U \\ B_{LU}\Delta u_U \end{bmatrix} = \begin{bmatrix} B_{UL}\Delta u_L \\ B_{LL}\Delta u_L \end{bmatrix}$$

Let us now define a control reallocation matrix $[\lambda]$ such that

$$[\Delta u_U] = [\lambda][\Delta u_L] \quad \Rightarrow \quad \begin{bmatrix} B_{UU} \\ B_{LU} \end{bmatrix}[\lambda] = \begin{bmatrix} B_{UL} \\ B_{LL} \end{bmatrix}$$

Define a linear relationship $[\alpha][\lambda] = [\beta]$

$$[\alpha][\lambda_1 \quad \lambda_2 \quad . \quad . \quad \lambda_m] = [\beta_1 \quad \beta_2 \quad . \quad . \quad \beta_m]$$

LP Formulation (Cont'd)

$$\min_{\lambda_i} \; (w_i{}^T \lambda_i)$$

Subject to

$$[\alpha][\lambda_i] \le [\beta_i] \quad \text{and} \quad 0 \le \lambda_i \le \lambda_{\max}$$

Example: 4 control inputs

$$[W] = \begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ w_{21} & w_{22} & w_{23} & w_{24} \\ w_{31} & w_{32} & w_{33} & w_{34} \\ w_{41} & w_{42} & w_{43} & w_{44} \end{bmatrix} = [w_1 \quad w_2 \quad w_3 \quad w_4]$$

Conventional & Best Hierarchies

	Elevator	Left Aileron	Right Aileron	Rudder
Elevator	**Primary**	***Secondary***	***Secondary***	
Left Aileron		**Primary**	***Secondary***	*Tertiary*
Right Aileron		***Secondary***	**Primary**	*Tertiary*
Rudder		***Secondary***	***Secondary***	**Primary**

Conventional

$$[W]^T = \begin{bmatrix} * & 1 & 1 & 100 \\ 100 & * & 1 & 10 \\ 100 & 1 & * & 10 \\ 100 & 1 & 1 & * \end{bmatrix}$$

Best

$$[W]^T = \begin{bmatrix} * & 1 & 1 & 100 \\ 100 & * & 1 & 1 \\ 100 & 1 & * & 1 \\ 100 & 1 & 1 & * \end{bmatrix}$$

Implementation

- Primary Cost based on “surface”

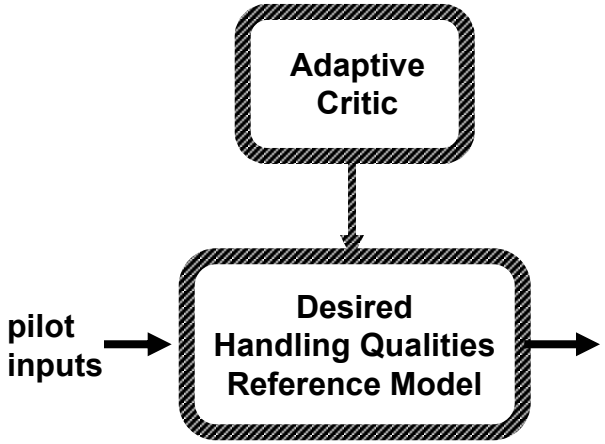
$$\min_{u} (w^T u)$$

- Auxiliary Cost based on “axis error”

$$\min_{u} (c^T e)$$

Level 2 Controller

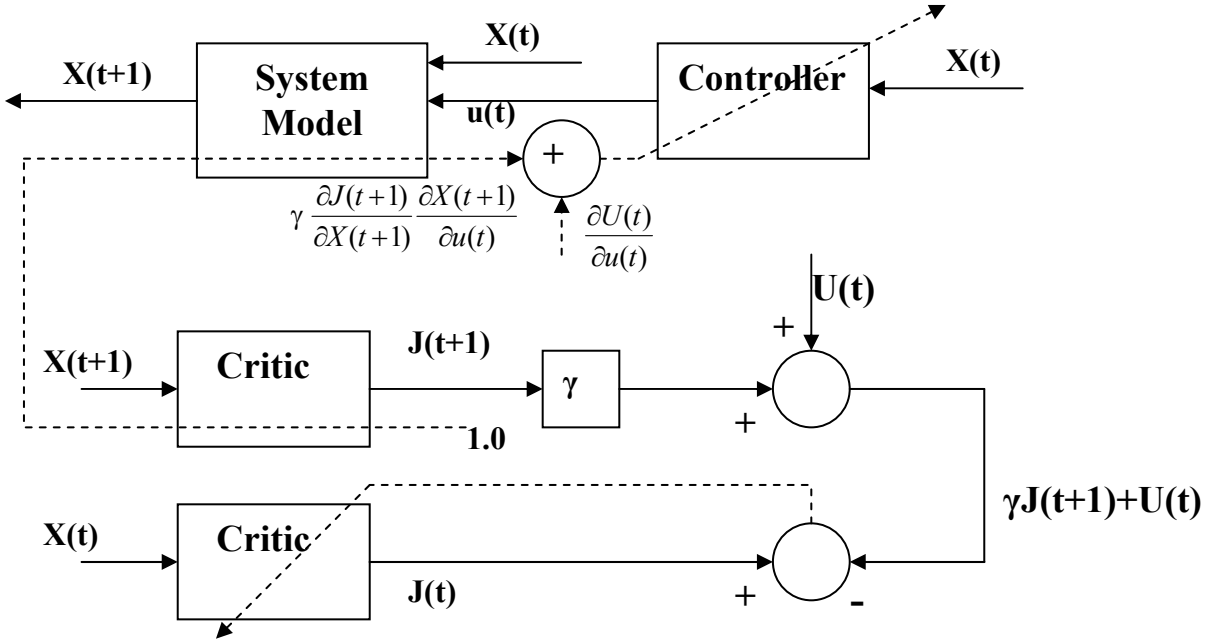
Reference Model Adaptation using an Adaptive Critic



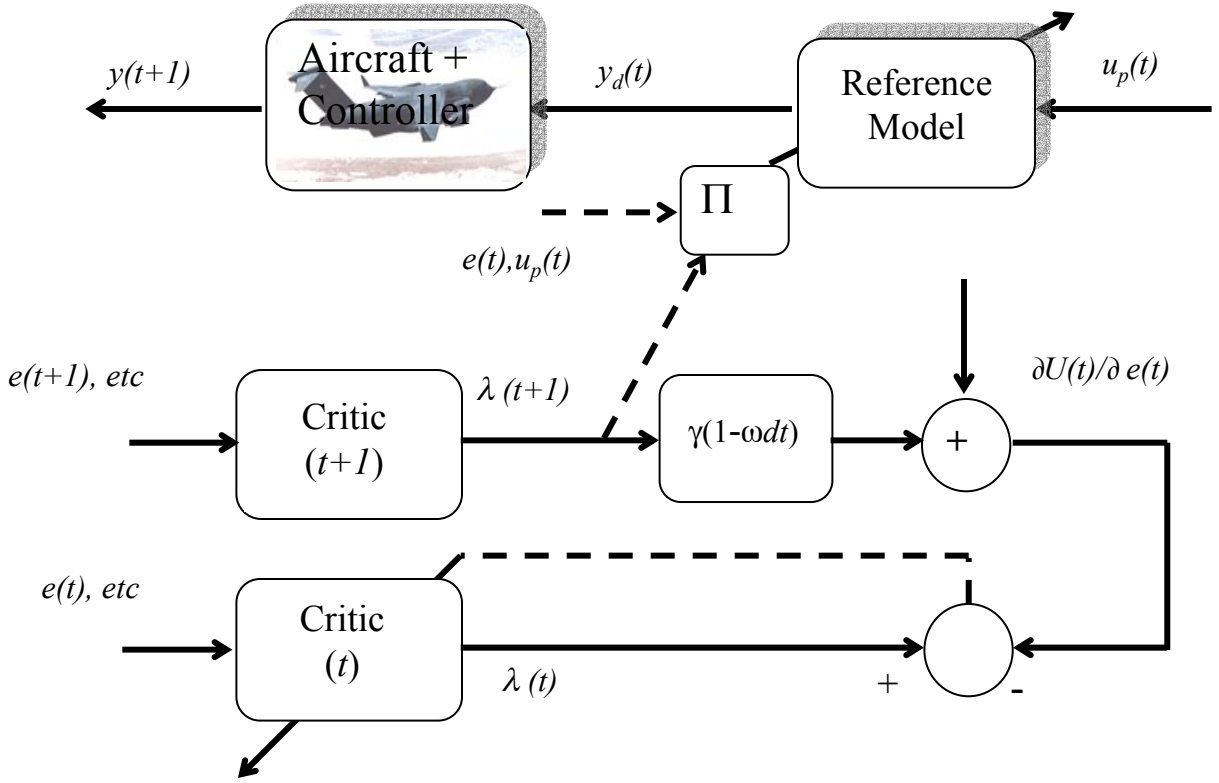
Adaptive Critic

Adaptive critic designs have been defined as designs that attempt to approximate dynamic programming.

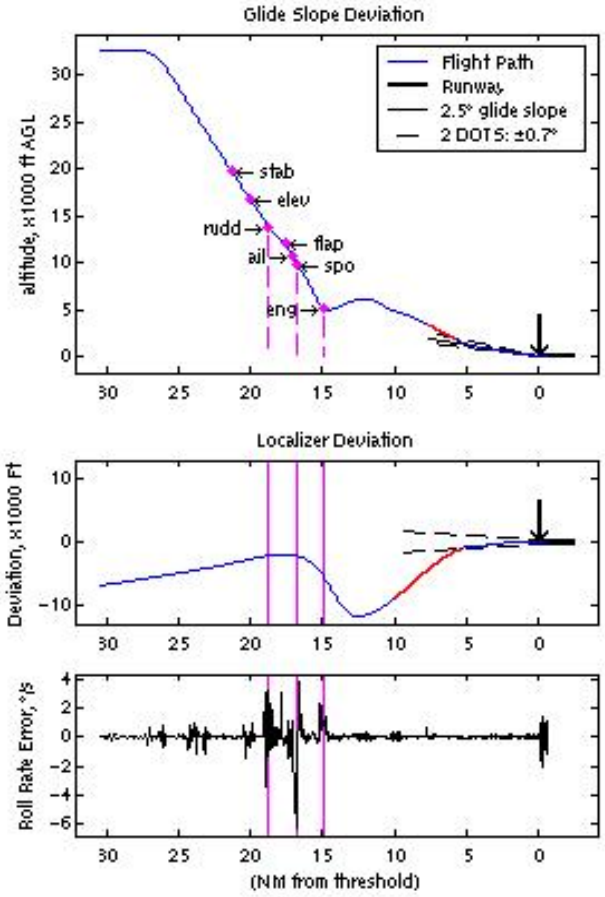
$$J(t) = < \gamma J(t+1) > + \min_{u} U(t)$$

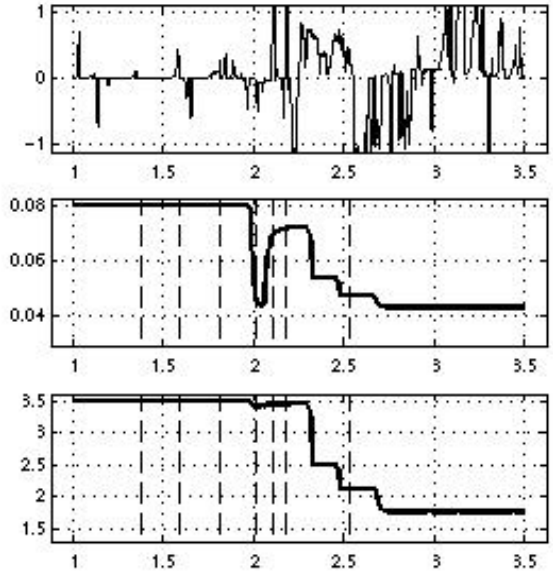


Level 2 Control



Results for Series of Failures





During tactical descent (failures on one side)

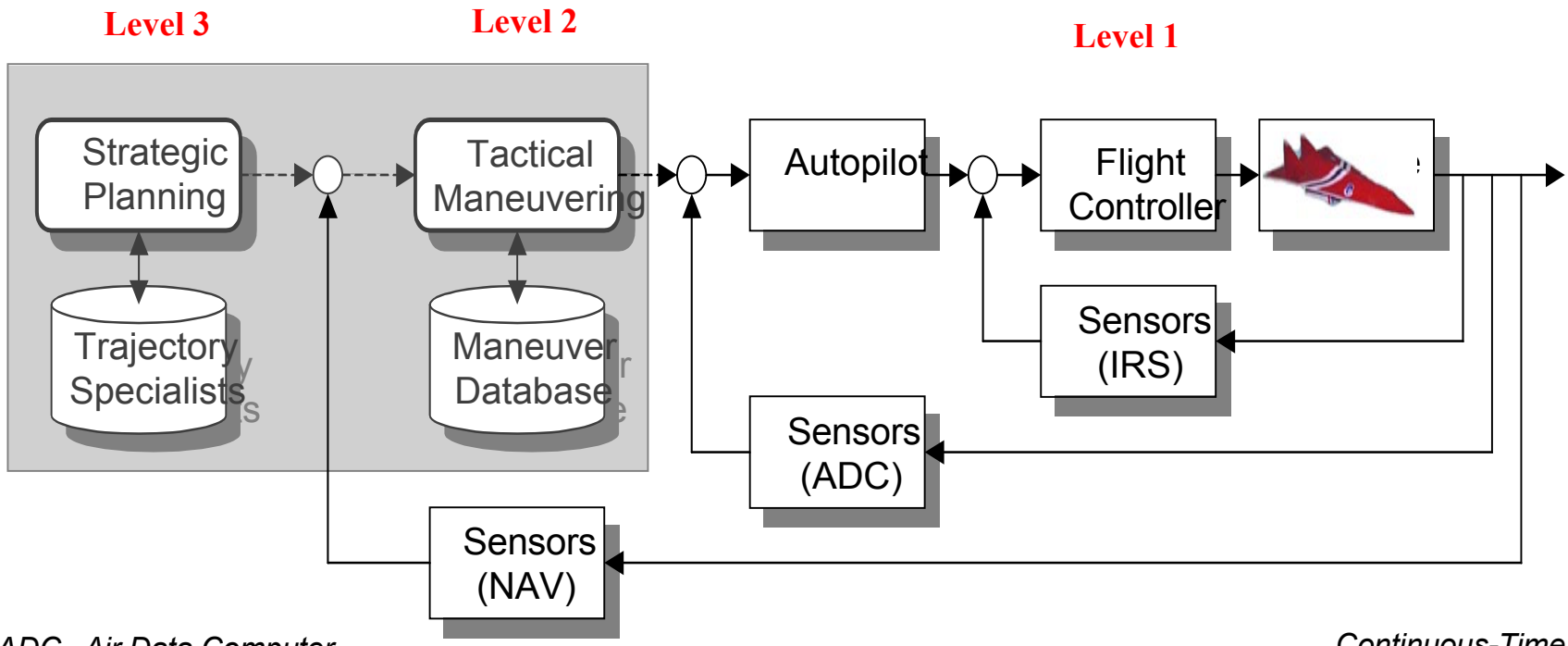
- 23,000': Stab frozen at trim
- 20,000': 2 Elevators frozen at 0 deg.
- 17,000': Upper rudder hard over
- 15,000': Outboard flap fails retracted
- 14,000': Aileron frozen at 0 deg.
- 13,000': Two outboard spoilers frozen at 0 deg.

When engines come out of reverse: Outboard engine seizes

Intelligent Maneuvering of UAVs

- *Goals*
  - ***Provide increasingly** higher levels of automation**, capable of responding to changing goals and objectives, while taking corrective actions in the presence of internal or external events.***
  - ***Allow pilots, ground-based operators or **autonomous executives** to defer the responsibilities of performing and supervising tasks, to focus on** managing goals and objectives*.

Intelligent Maneuvering of UAVs

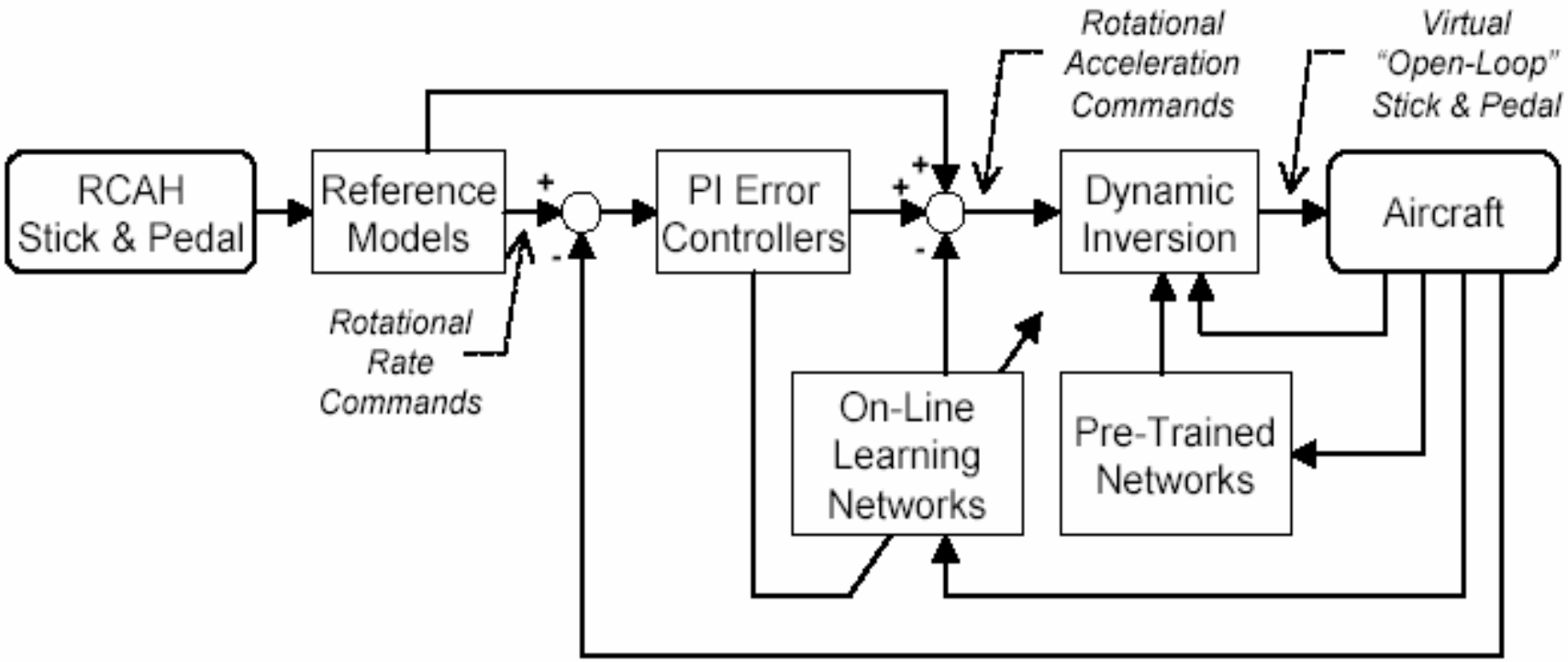


ADC- Air Data Computer
IRS - Inertial Reference System
NAV- Navigational System (GPS & Visual Perception)

Continuous-Time Commands & Sign

Discrete-Time Commands

Flight Controller

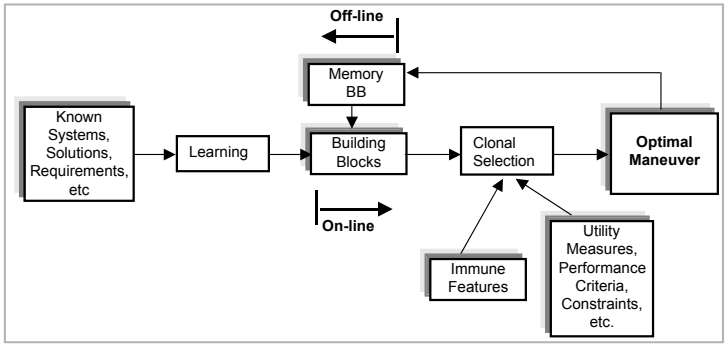


Tactical Maneuvering

Performs time-critical flight path operations, which includes aggressive maneuvers in the presence of unexpected obstacles.

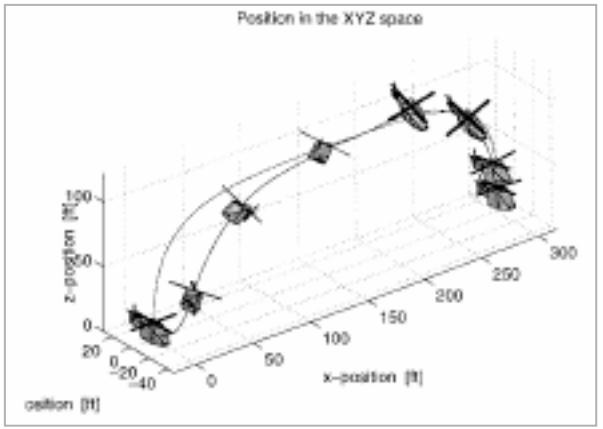
- Inputs
  - –Commands
    - •Reference Targets / Trajectory
    - •Performance Parameters
  - –Awareness
    - •Threat Detection *(eg. TCAS, GCAS)*
    - •Vehicle Performance Models
- •Outputs
  - –Maneuver Sequence
    - •Control Law Specific Modes & Targets
    - •Transition Criteria
- •Maneuver Selection Specialists
  - **–Immunized Maneuver Selection**
  - –Heuristic-Based TSP Maneuver Selection
- •Maneuver Database
  - –Elements & "Canned" Sequences

Immunized Maneuver Selection



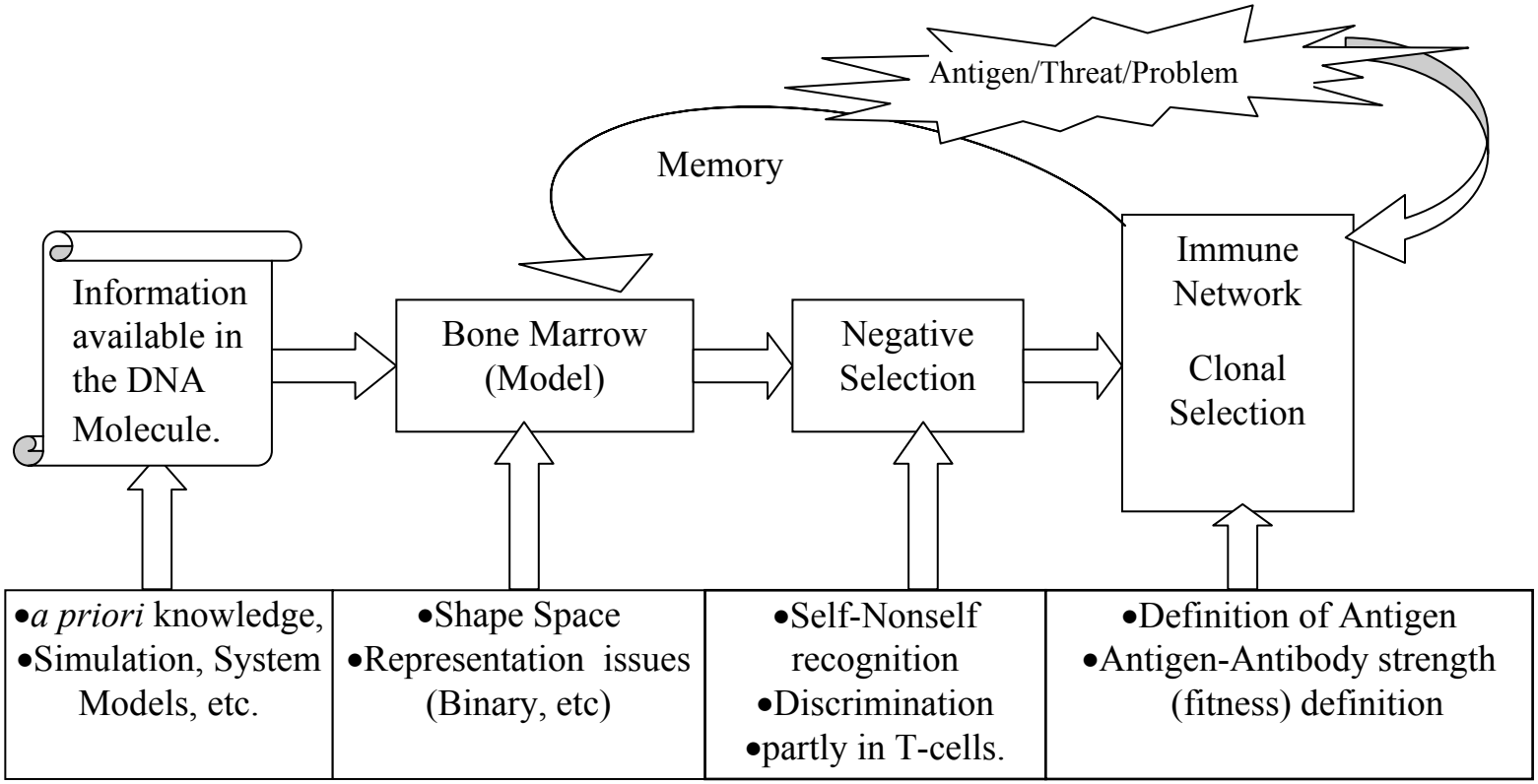
Krishna Kumar

Model Predictive Take-off and Landing



Eric Wan

A system-level description of the Immune System Metaphor



Tactical Maneuvering Database

Contains general and aircraft specific maneuvering database elements, each corresponding to associated control laws. Pre-canned maneuver sequences represent domain expertise.

- Elements
  - –Control Law
    - •Mode & Target Definition
  - –Aircraft Specific
    - •Flight Envelope Validation Logic
  - –Specifications
    - •Closed-Loop Predictive Models *(x_0, …, x_f)*
    - •Resource Allocation Table *(e.g. lat, lon, ped, thr/col)*
- Sequences
  - –Elements
    - •Specified Parameters / Arguments
  - –Transition Criteria / Termination Logic
    - •Time-Based and/or Condition-Based
  - –Interrupts
    - •Abort Conditions & Abort Sequence

Bank to Turn Element

Heading Select (coord. turn)

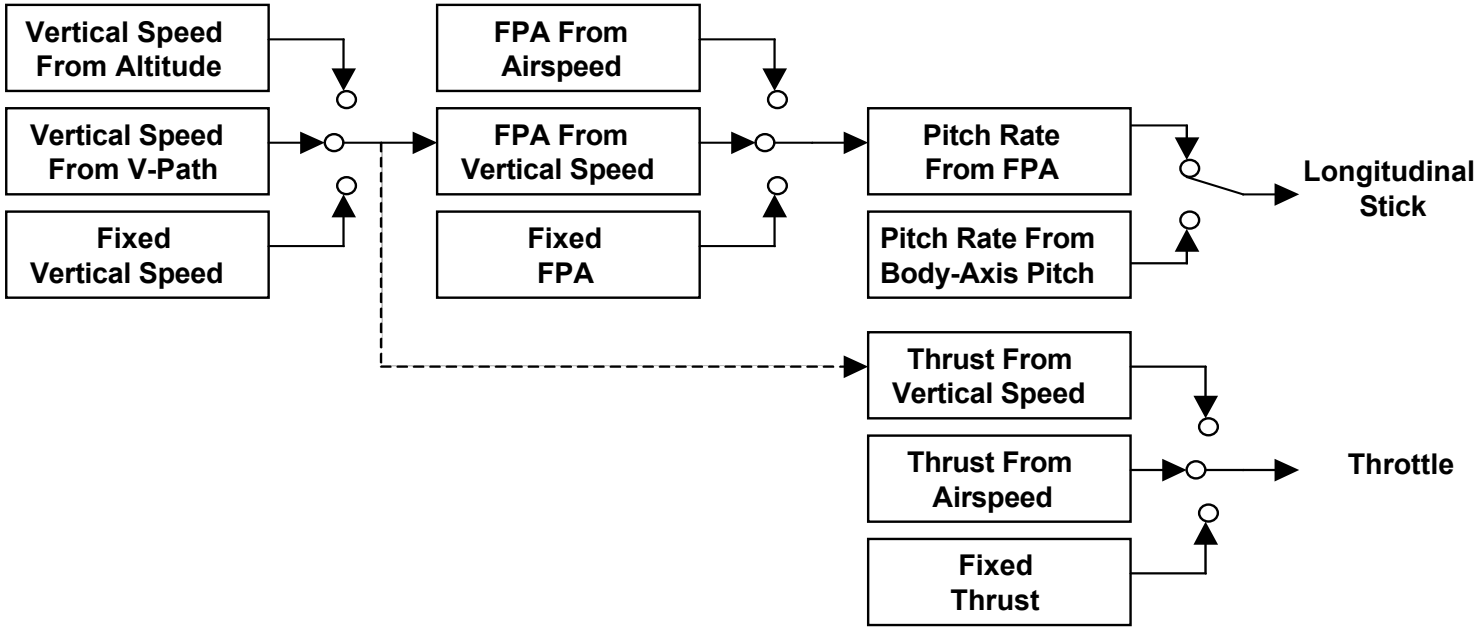
Mode: HDGSEL
Target: Heading = [arg1] deg
Envelope: IAS > 180 kts, $|\theta| < \ldots$
Model: $\varphi_o/\varphi_i = \tau/(\tau s+1)$,
$\varphi'_{max} = g\cos(\theta)\sin(\phi_{max})/v_t$
RAT: LAT/PED

Bank & Pull to Turn Sequence

Bank Left: 0
Normal Accel. (speed control)
Bank Right: +90

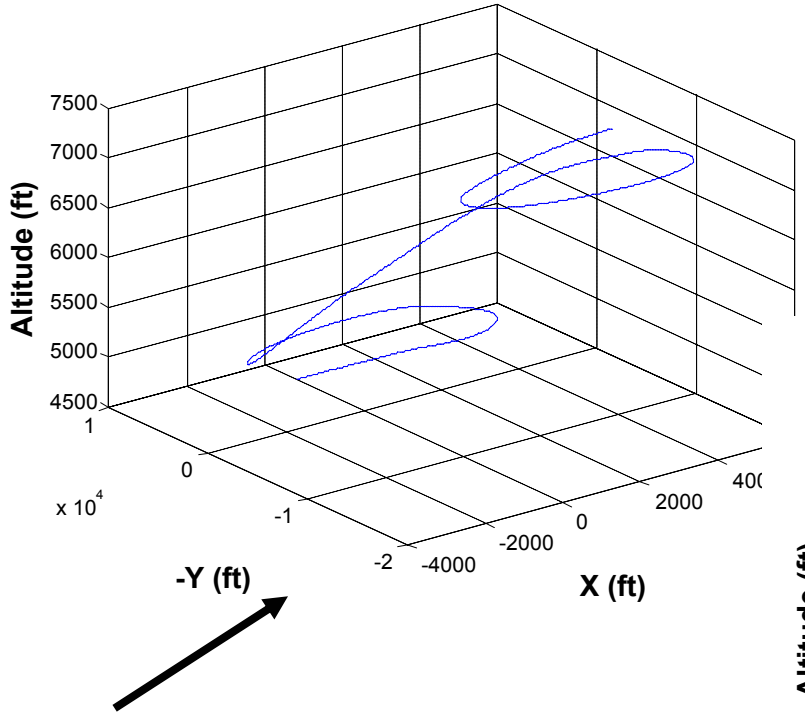
Mode: BANKSEL
Target: Bank = 90 deg, $Vz = Vz_0$
Envelope: mach > 0.4, $|\alpha| < \ldots$
Model: $\phi_o/\phi_i = \tau/(\tau s+1)$,
$\phi'_{max} = p_{max}$
$Vz = Vz_0$
RAT: LAT/PED

Autopilot System (Example)

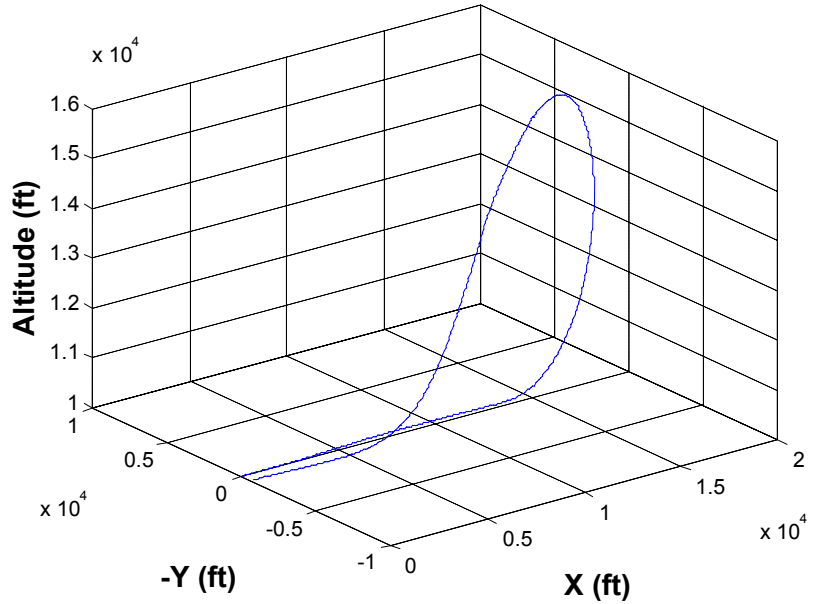


- Longitudinal Modes
  - –Pitch, Nz, AoA, FPA; Mach, IAS, Vertical Speed; Vertical Path, Altitude
- Thrust Modes
  - –Mach, IAS, Vspd, Thrust; Vertical Path, Altitude, FPA
- Lateral Modes
  - –Bank, Roll Rate; Heading, Track; Lateral Path
- Directional Modes
  - –Sideslip, Ny, Heading

Results



Three Chained Modes →

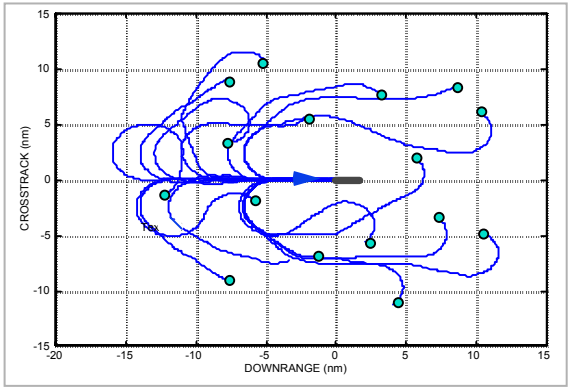


Strategic Maneuvering

Performs long-term planning that meets dynamic mission goals and objectives, within mission constraints and performance limitations.

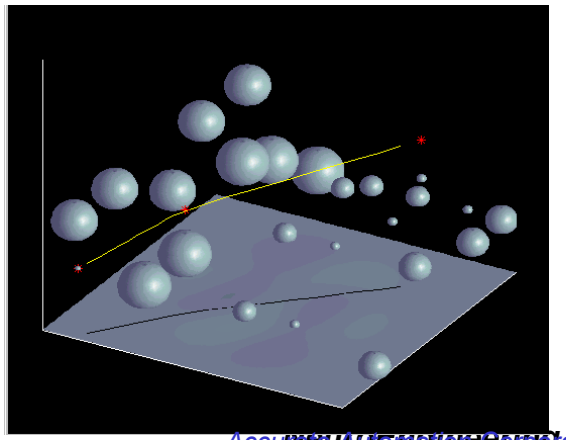
- Inputs
  - –Goals
    - •Cost Function
    - •Mission Constraints
  - –Awareness
    - •External Obstacles *(weather, terrain, ...)*
    - •Internal Health & Performance Limitations
- Outputs
  - –Extended Flight Plan
    - •Waypoints / Reference Trajectory
    - •Performance Parameters
    - •Configuration Schedules
- Trajectory Specialists
  - –Energy Management Guidance
    - •Tear Drop, Low-Altitude, Enroute
  - –Evolutionary Navigation

Low-Altitude Energy Management

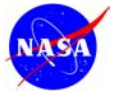


John Bull

Obstacle Avoiding Evolutionary Navigation



Optimal Way Point Computation Around Obstacles Using Evolutionary Algorithms

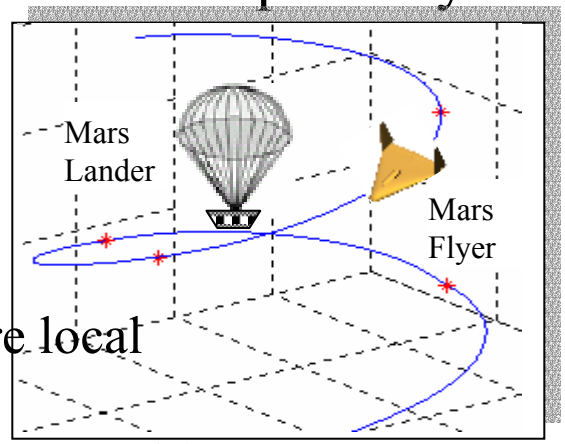


- **The Algorithm:**
  - Step 1: Determine the obstacles that are in the path of the flight
  - Step 2: Place the waypoints for the aircraft on the circumference of the obstacles
  - Step 3: Compute the path between the start and the end using the waypoints.
  - Step 4: Compute a fitness function
  - Step 5. After "n" iterations the best set of waypoints defines the navigation path.

Demo

Intelligent Control for BEES

Exploration of Mars using Free-flyers with sensors inspired by Nature



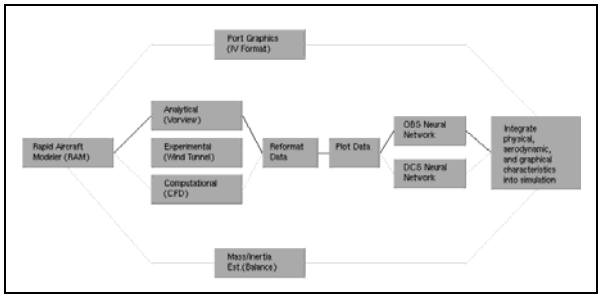
Controller Objectives:

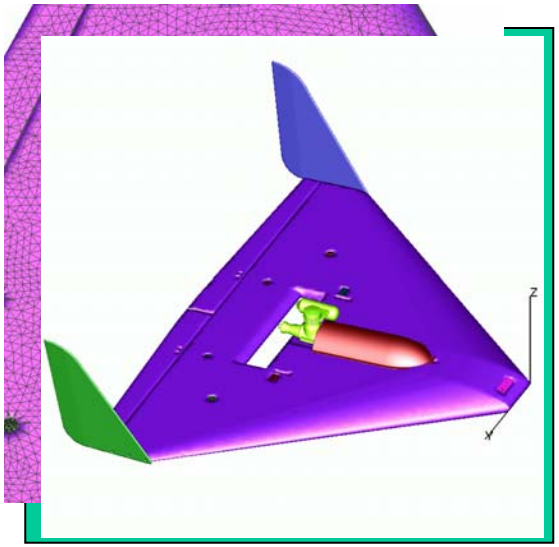
- Maintain safe distance from the **Lander** and ensure local stability.
- Point in the desired attitude and follow a trajectory to enable imaging of interesting Geological Picture.
- Optimize long-term and short-term goals, such as minimization of fuel (long-term) and avoid collision with the **Lander** (short-term)
- React to changing environments by adapting the control functionality

Pre-Trained Neural Networks

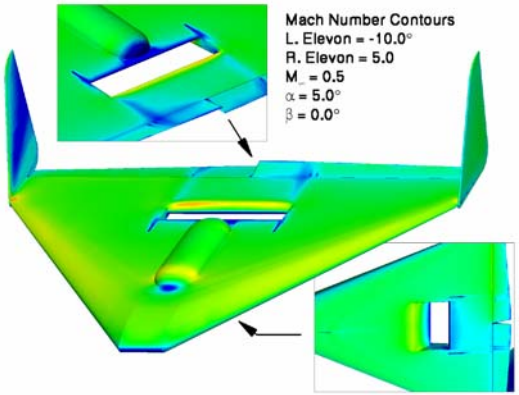
Ames Research Center

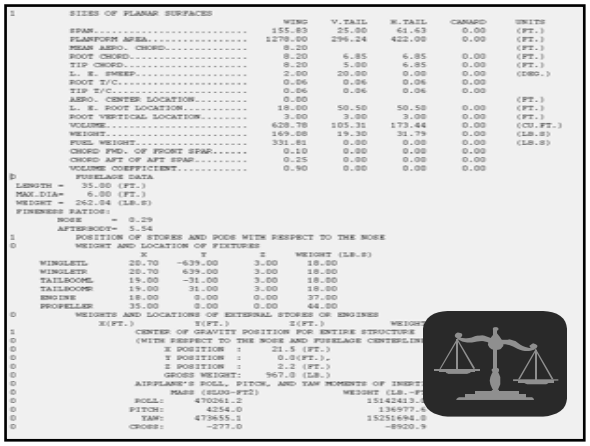
Step 1
Integrated Vehicle Modeling Env.
CAD Designs



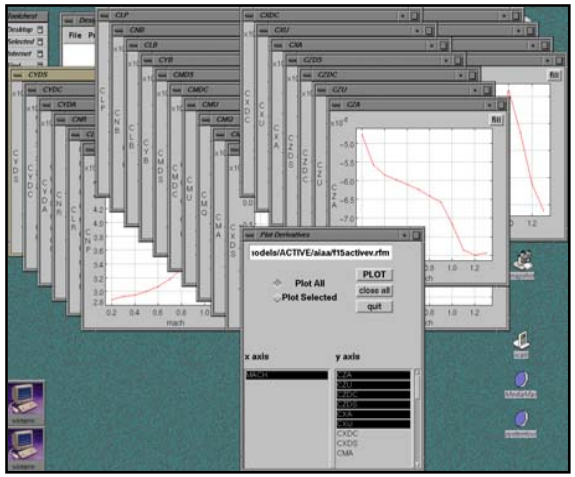


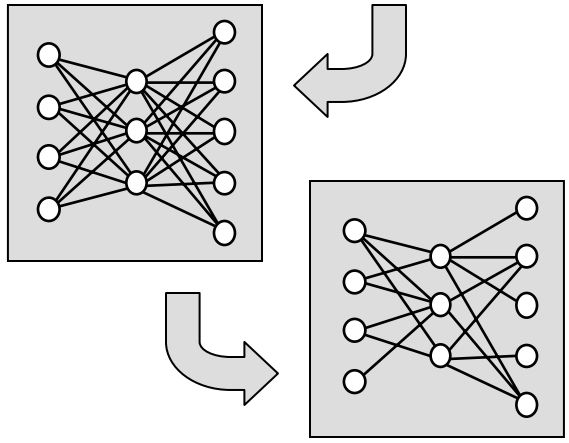
Step 2
Cartesian Euler Code (CART3D)



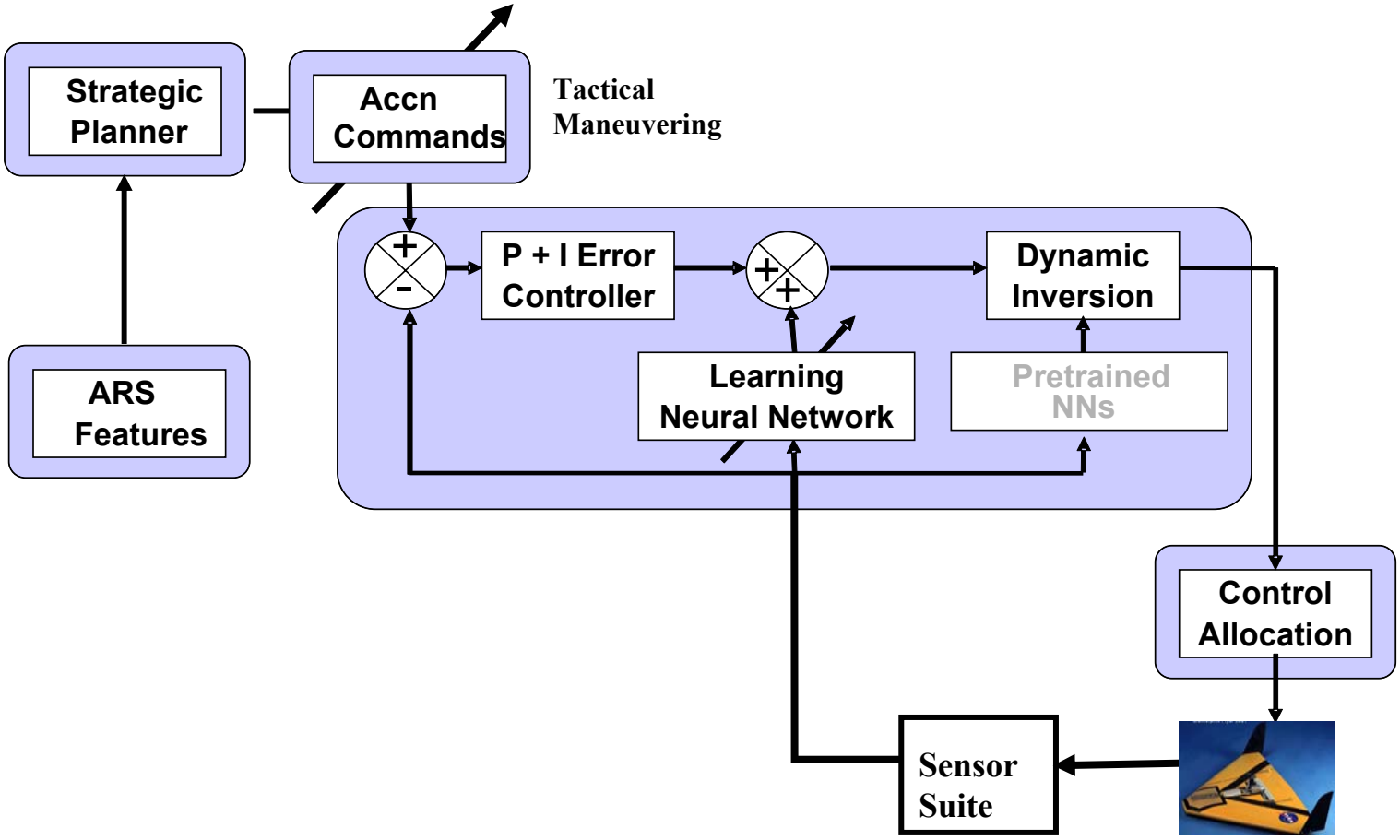


Step 3
Levenberg Marquardt neural net Optimal Pruning Algorithm





Level 2 Architecture



Concluding Remarks

✓Intelligent control comes in many flavors
✓Levels of Intelligent Control is one way of quantifying the roles of Intelligent control
✓Intelligent control architectures allow for fast prototyping
✓Intelligent control architectures can guarantee inner-loop stability
✓For UAV application, intelligent control provides a robust way to accommodate any outer-loop architecture for planning, etc.